Разбор заданий окружного этапа
Всероссийской олимпиады школьников по информатике
2013 – 2014 уч. год, Самара

I тур

Задача A. Ассоциативное мышление.

В данных ограничениях достаточно было аккуратно симулировать процесс.
В качестве k можно оставить $k = k\, mod\,(2 \cdot n - 2)$ (исключая полные круги, описываемые профессором). В логическом массиве соответствующей длины фиксировать факт, что лавочка уже связана с каким-то открытием. Как только профессор вновь окажется возле одной из уже отмеченных лавочек, он отправится домой. Количество совершённых им открытий будет при этом равно количеству ассоциированных с лавочками плюс одно – то, которое он не смог связать ни с какой лавочкой.

Задача B. Точка, точка, запятая...

В этой задаче также требовалось аккуратно обработать данный текст. Можно было воспользоваться “идеологией” конечного автомата: определить состояния (“прочитана буква”, “прочитан пробел”, “прочитан знак препинания” – тут следовало отдельно рассматривать дефис, тире и точку, поскольку она может быть частью многоточия), определить допустимые переходы, и определить, как следует корректировать текст, если переход недопустим. Поскольку сам текст в ответе выводить не требовалось, то вполне достаточно было завести счетчик символов, изначально равный количеству элементов в тексте, а затем увеличивать или уменьшать его в зависимости от того, какое необходимо внести исправление.

Задача C. В поисках непротиворечивости.

Существует несколько подходов к решению этой задачи. Опишем здесь два.
Первый подход, состоит в том, что мы попробуем как-либо упорядочить переменные по возрастанию. Для каждого неравенства вида $x_i > x_{j1} + x_{j2} + \ldots + x_{jk}$ запишем $x_i > x_{j1}$, $x_i > x_{j2}, \ldots, x_i > x_{jk}$; для каждого неравенства вида $x_i \ge x_{j1} + x_{j2} + \ldots + x_{jk}$ запишем $x_i \ge x_{j1}$, $x_i \ge x_{j2}, \ldots, x_i \ge x_{jk}$.Следует учитывать, что если найдётся пара неравенств вида $x_{p1} \ge x_{p2}$ и $x_{p2} \ge x_{p1}$, то это означает, что $x_{p1} = x_{p2}$. В этом случае нужно везде заменить одну переменную другой (например, x_{p2} на x_{p1}) и повторить описанную выше процедуру. Если встретили противоречие (например, $x_{q1} > x_{q2}$ и при этом $x_{q2} > x_{q1}$), то систему считаем неразрешимой.
Второй подход реализационно несколько проще. Положим все переменные равными единице. Много раз повторим следующее: найдём какое-нибудь неравенство, выполняющееся неверно при текущем наборе значений x, после чего минимально увеличим переменную в леовй части этого неравенства так, чтобы оно выполнялось.
Основываясь решении, описанном в первом подходе, можно показать, что таким образом мы найдём ответ, если он существует. Число шагов, которое для этого потребуется, зависит от конкретной реализации. Если число шагов превышает некоторое пороговое значение, то система объявляется неразрешимой. Определить пороговое значение можно из ограничений задачи.

Задача D. Надпись на стене.

Будем рассматривать подстроки длины 2^j, начинающиеся с позиций $i=1..n-2^j+1$, для $j=0..k$. Мы хотим каждой такой подстроке поставить в соответствие некоторое число, причем таким образом, чтобы лексикографически меньшей строке соответствовало меньшее число, а равным строкам соответствовали равноые числа.

Начнем с $j=0$. Легко составить такое соответствие по $s[i]$: $'a'=1$, $'b'=2$, $'c'=3$, $\ldots,'z'=26$.

Пусть теперь нам известен набор $number[i]$ для некоторого j. Как можно построить набор для $j+1$?

Подстрока длины 2^{j+1} может быть представлена как конкатенация двух подстрок длины 2^j. Чтобы сравнить две строки длины 2^{j+1} на позициях i_1 и i_2 нам достаточно сравнить пары $(number[i_1],number[i_1+2^j])$ и $(number[i_2],number[i_2+2^j])$. Выпишем такие пары для $i=1..n-2^{j+1}+1$, упорядочим и каждой из них назначим новый номер. Он и будет подходить для подстрок длины 2^{j+1}.

Задача E. Весомые аргументы.

Как легко заметить, в этой задаче нужно было просуммировать введённую числовую последовательность и сравнить полученный результат с *p*.

II тур

Задача A. Продуктовая корзина.

В текущих ограничениях можно было решать задачу непосредственным подсчётом. Понятно, что все товары нам придется обязательно взвесить. Поэтому можно вычислить математическое ожидание времени, затраченное на переходы между весами, а потом просто добавить к нему *n*.

Для вычисления времени переходов можно действовать так: рассматривать последовательность товаров как состоящую из групп, которые можно подряд взвешивать на одних и тех же весах. Понятно, что в “минимальном” варианте будет одна группа для одних весов и другая группа для других весов. Такое количество групп требует только одного перехода между весами. Далее количество групп можно увеличивать (притом количество групп товаров для разных весов не может отличаться более, чем на единицу).

Вычислить количество групп можно с помощью биномиальных коэффициентов, однако обходиться с ними следует аккуратно: например, C_{100}^{50} не поместится даже в тип *long*.

Впрочем, есть более элегантный способ. Рассмотрим последовательность товаров как последовательность “событий” – переходим мы к другим весам перед взвешиванием этого товара или не переходим. Число перемещений – это число перемещений перед 1 взвешиванием + число перемещений перед 2 взвешиванием + ... + число перемещений перед *n*-ым взвешиванием. Несмотря на то, что эти случайные величины нельзя считать независимыми, математическое ожидание их суммы вычисляется как сумма математических ожиданий каждой величины в отдельности.

Среднее число перемещений перед товаром 1 – это вероятность этого перемещения (поскольку считается как сумма: 0 *<вероятность не переместиться> + 1 * <вероятность переместиться>). Что касается вероятности того, что на позиции *i* будет товар, взвешиваемый на первых весах, а на позиции *i*+1 – товар, взвешиваемый на вторых весах (или наоборот), то она вычисляется непосредственно. Действительно, существует всего C_n^2 возможностей для этих двух товаров, из них *a*b* являются подходящими. Просуммировав все, получаем

$n+0.5+(n-1)\cdot 2\cdot a\cdot b/n/(n-1)=n+0.5+2\cdot a\cdot b/n$, где 0.5 возникает за счет выбора весов при взвешивании первого товара из корзинки.

Задача B. Зеленый и черный.

В этой задаче нужно было последовательно обрабатывать символы строки и поддерживать количество чашечек чёрного и зелёного чая, выпитых профессором. Разумеется, нужно было учитывать, что когда профессор пьёт чашечку зелёного чая, заваренного им лично, чай не перестаёт от этого быть зелёным (и количество выпитых чашечек зелёного чая увеличивается, а количество выпитых чашечек чёрного чая не меняется).

Задача C. Восстановление пути.

Эту задачу можно разбить на несколько подзадач.

1) Ищем кратчайшие пути от 0. Вычищаем граф, оставляя только ребра, для которых $d[u]+1=d[v]$

2) Разбиваем все скульптуры на группы с одинаковым кратчайшим расстоянием. Если в такой группе обнаруживается две (и более) пометки – профессор ошибается, нужно выводить BAD.

3) Определяем для каждой скульптуры, можем ли мы добраться по “вычищенному” графу до следующей (по расстоянию) метки или, если такой нет, до дома какого-то друга. Это делаем, рассматривая группы по убыванию расстояния от 0. Попутно определяем BAD-ситуации, связанные с тем, что через саму дальнюю метку не проходит ни один кратчайший путь от дома какого-либо друга.

4) Находим для каждой скульптуры, можем ли мы добраться до неё по “вычищенному” графу от предыдущей (по расстоянию) пометки или, если такой нет, от 0. Это делаем, рассматривая группы по возрастанию расстояния от 0. Попутно определяем BAD-ситуации, связанные с тем, что какая-то метка не лежит на кратчайшем пути от другой метки до вершины 0.

5) Скульптура может иметь пометку, если для неё в 3) и 4) были получены ответы “да”.

Задача D. Грант

Рассмотрим маршрут - ответ. Обозначим через $x[i]$ количество раз, которое профессор проходил по *i*-ой дорожке (суммарно вперёд и назад). Пусть теперь профессор посетил некоторый корпус $2k+1$ раз. Тогда он $2k+1$ раз приходил в этот корпус и $2k+1$ раз из него выходил, т.е. если $i1, i2, \ldots$ – дорожки, по которым можно попасть в этот корпус, то $x[i1]+x[i2]+\ldots=4k+2$, т.е. $x[i1]+x[i2]+\ldots=2(mod\, 4)$.

Найдем некоторое решение этой системы уравнений. Протопчем новые дорожки так, чтобы, например, если $x[i]=3$, то стало три дорожки, соединяющие корпуса, которые соединяет сейчас дорожка *i* (включая *i*). Если $x[i]=0$, то соответствующую дорожку засадим газоном. Теперь из каждого корпуса выходит четное число дорожек, значит, как только по нашим дорожкам можно будет дойти из любого корпуса в любой, будет существовать маршрут, проходящий все дорожки ровно по одному разу. Найдем набор из $n-1$ дорожки, связывающий все корпуса до того, как мы начали наши преобразования (например, так называемым поиском в глубину). Добавим каждую дорожку этого маршрута четырежды, так мы не нарушим условия системы уравнений.

Поиск такого маршрута (он называется эйлеров цикл) осуществляется следующим способом. Начнем с корпуса 1. Дальше на каждой итерации мы либо берем любую дорожку, которую еще не проходили, и кладем ее в стек, либо, если такой дорожки нет, забираем из стека верхнюю дорожку и записываем ее в ответ. При этом корпус, из которого выходят рассматриваемые дорожки, переходит на конец и начало дорожки (соответственно). Более подробное обсуждение этого алгоритма можно найти в интернете.

Теперь, когда мы поняли, что решение этой системы уравнений позволит решить задачу, осталось эту систему решить! Будем действовать способом, аналогичным методу Гаусса. (Подробно о методе (исключений) Гаусса можно прочитать в различных источниках, в том числе в интернете; далее предполагается, что читатель с ним знаком). Решение будем проводить в несколько этапов. По модулю 4 мы не можем поделить на 2 (а на 3 можем, так как $3*3=1 mod 2$, т. е. чтобы поделить на 3, надо просто помножить на 3), поэтому на первом этапе мы никогда не будем выбирать в качестве ведущего элемент со значением 2. В итоге у нас останется матрица, в которой вне строк, где уже выбраны ведущие элементы, все числа чётные. Продолжим метод, теперь выбирая и 2 как возможный ведущий элемент. Нам не придётся делить на 2, чтобы работать с этими строками; правда, мы не сможем исключить такую переменную из строк, в которых ведущий уже выбран, как, например, в случае, описанном ниже. В итоге после преобразований у нас останется такая матрица, что в строках, где ведущий элемент 2, он единственный (не считая свободных). В примере ниже становится ясно, что мы можем произвольно выбирать значение такой переменной с точностью до четности (или система неразрешима: например, $2x=1$). Примем любую из двух возможностей, заменив соответствующее уравнение в системе. Теперь соответствующий элемент равен 1, и мы можем продолжать исключения до момента, когда в каждом уравнении останутся только ведущие элементы и свободные переменные. Свободные переменные примем за 0 и найдем таким образом решение системы.

Например, система

$x1+x2+2*x3=1$

$x1+3*x2=3$

после первого этапа

$x1+x2+x3=1$

$2*x2+2*x3=2$

после второго этапа

$x1+x2+x3=1$

$x2+x3=1$

после исключений

$x1=0$

$x2+x3=1$

В итоге, например, $x1=0, x2=1, x3=0$ является решением.

Еще один подход решению заключается в следующем. Сформулируем несколько утверждений.

Утверждение 1. Если $m=n-1$, то, если *n* нечетно, любой обход, посещающий нечетное число раз все вершины, кроме 1, посещает вершину 1 четное число раз.

Утверждение 2. Если $m=n-1$, то, если *n* четно, любой обход, посещающий нечетное число раз все вершины, кроме 1, посещает вершину 1 нечетное число раз.

Утверждение 3. Если профессор не может выйти из какого-то корпуса и вернуться в него же, посетив при этом нечетное количество корпусов, и n нечетно, то задача неразрешима.

Все три утверждения могут быть доказаны следующим способом. Раскрасим корпуса в черный и белый цвет, причём так, чтобы два корпуса, соединённые тропинкой, имели разные цвета. В случаях, указанных в утверждениях выше, это возможно. Тогда "маршрут" профессора будет по очереди проходить черные и белые корпуса. Число посещений, а, значит, и чётность числа посещений черных и белых корпусов будет совпадать. Во всех трех утверждениях мы от противного получаем противоречие этому условию.

Случай 1. Число *n* четно. Будем обходить корпуса, например, поиском в глубину. Если, когда мы собираемся уйти из вершины, число ее посещений чётно, “зайдём” в её предка, вернёмся назад, и пойдём дальше. Все корпуса, кроме начального, мы посетили нечётное число раз и при этом ходили только по *n-1* ребру. Значит, число посещений начального корпуса также будет нечётно (утверждение 1).

Случай 2. Число *n* нечётно. Найдём маршрут, не позволяющий нам покрасить все корпуса в два цвета. Если такого не существует, решения нет. Если он существует, будем действовать как в случае 1, начиная из какого-нибудь корпуса на найденном маршруте. Теперь оказалось, что все корпуса посещены нечётное количество раз, кроме того, из которого мы начинали поиск. Пусть найденный маршрут, например, 1 2 3 4 5 1, а поиск мы начинали с корпуса 1. Пойдём следующим образом: 1 2 3 2 3 4 5 4 5 1 (два шага вперёд – шаг назад – шаг вперёд – два шага вперёд – ...). Легко заметить, что мы посетим чётное число раз все корпуса кроме того, с которого начали обход, т.е. теперь все корпуса суммарно посещены нечётное число раз.

Задача E. Вечер трудного дня.

В этой задаче нужно было аккуратно рассмотреть четыре возможности. Первая и вторая – когда дно одного стакана может быть “совмещено” с дном другого. В этом случае достаточно вычислить максимальную из высот стаканов. Третья и четвёртая состоят в том, что один стакан помещается в другой так, что до дна недостаёт. Из соображений подобия можно найти, какая часть высоты при этом “поглощается”.